Active Project (2014 - 2017)

A Scalable Semantics-Based Verification System for Flight Critical Software, Phase II Project

SBIR/STTR Programs | Space Technology Mission Directorate (STMD)

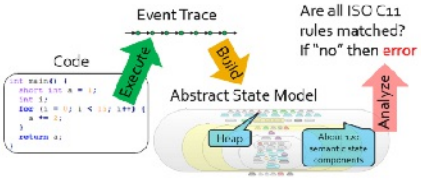
ABSTRACT

Flight-critical systems rely on an ever increasing amount of software—the Boe- ing 777 contains over 2 million lines of code. Most of this code is written in the C programming language. We need a scalable static formal program verification tool that is able to prove the functional correctness of flight-critical software, limiting any failure of flight critical software to hardware faults. This project seeks to leverage the matching logic verification framework. Matching logic is generic in an operational semantic of a given programming language, so we also seek to give a semantics of a subset of C, called CIL, which is guaranteed to be deterministic. While we already have a semantics for the entirety of C, CIL is more representative of flight-critical software, and the simpler, deterministic semantics will result in a more efficient, and thus more scalable, static program verification tool. We are also building a new unification- based rewrite engine that will result in a more powerful version of the Matching Logic Framework. In order to make the tool more commercially feasible, we will develop new techniques in pattern inference, so that loop invariants and some pre/post conditions can be determined automatically. We will perform a thorough evaluation of our tool on a large-scale piece of software with similar characteristics to a flight system.

ANTICIPATED BENEFITS

To NASA funded missions:

Potential NASA Commercial Applications: We anticipate that our tool and definition of CIL will be used in verification of all safety and mission critical flight systems. NASA has shown themselves to be very amenable to formal methods techniques in the past. Our technical contact within NASA expects such a tool to be used on vehicles targeted at manned space missions as well as mission critical systems such as the rockets, probes, space telescopes, and landers/rovers.

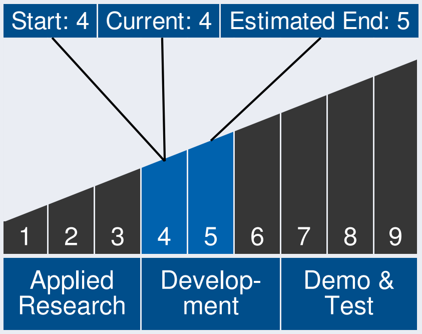

Event Trace
Are all ISO C11 rules matched? If "no" then error
Code
Execute
Build
Abstract State Model
Analyze
Heap
About [illegible] semantic state components

```
int main() {
  short int a = 1;
  int i;
  for (i = 0; i < 15; i++) {
    a += 2;
  }
  return a;
}
```

A Scalable Semantics-Based Verification System for Flight Critical Software, Phase II

Table of Contents

Technology Maturity



Management Team

Program Executives:

- Joseph Grant
- Laguduva Kubendran

Continued on following page.

To the commercial space industry:

Potential Non-NASA Commercial Applications: TAM for our work would be the entire software industry. Realistically, we do not have the manpower to target the entirety of the software industry. A more reasonable SAM would be those companies in the software industry where correctness is key, either because bugs are very expensive to fix (Microsoft) or where bugs are disastrous to life and limb (NASA/Boeing/medical device manufacturers). We feel that our realistic SOM is larger software development corporations that would be willing to pay larger amounts for site licenses as well as be willing to hire on verification engineers from our Runtime Verification, Inc. This will allow us to leverage our smaller number of employees in a way that results in maximal possible profits, versus marketing to smaller software developers, which will be less willing to pay for site licenses. We can move into that segment of the market later, once Runtime Verification is able to grow. Initially, we would like to focus on Aerospace companies (Boeing), Automotive (Toyota-ITC, DENSO), Finance (2Sigma), Microsoft (in particularly Windows and Office), Apple (OS X), the US Military, NSA, the large hadron collider, and Facebook (they have already shown themselves amenable to formal methods tools).

Management Team *(cont.)*

Program Manager:
- Carlos Torrez

Project Manager:
- Alwyn Goodloe

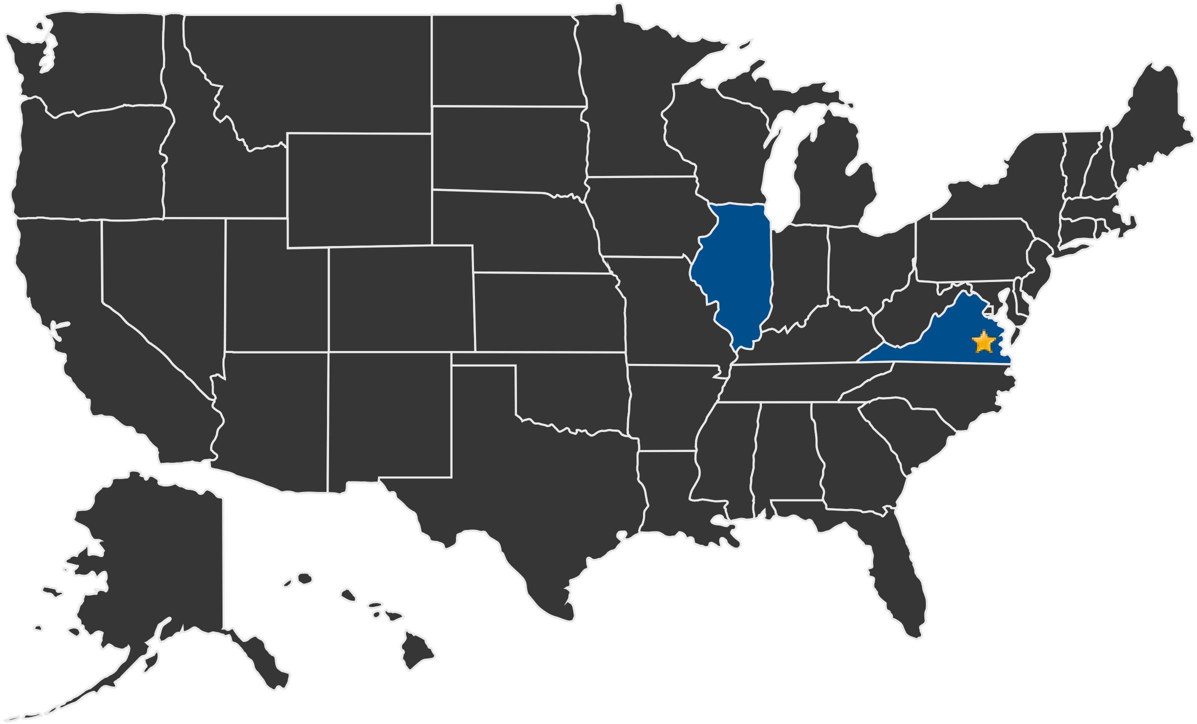
Principal Investigator:
- Dwight Guth

Technology Areas

Primary Technology Area:
Aeronautics (TA 15)
└ Enable Assured Machine Autonomy for Aviation (TA 15.6)
  └ Ability to Fully Certify and Trust Autonomous Systems for NAS Operations (TA 15.6.2)
    └ Assurance of Flight Critical Systems (TA 15.6.2.1)

Secondary Technology Area:
Robotics and Autonomous Systems (TA 4)
└ Systems Engineering (TA 4.7)
  └ Robot Software (TA 4.7.4)

U.S. WORK LOCATIONS AND KEY PARTNERS



🟦 U.S. States With Work

⭐ **Lead Center:**
Langley Research Center

Other Organizations Performing Work:

- Runtime Verification, Inc. (Champaign, IL)

PROJECT LIBRARY

Presentations

- Briefing Chart
  - (http://techport.nasa.gov:80/file/23078)
- Final Summary Chart
  - (http://techport.nasa.gov:80/file/23809)

IMAGE GALLERY

A Scalable Semantics-Based Verification System for Flight Critical Software, Phase II

DETAILS FOR TECHNOLOGY 1

Technology Title

A Scalable Semantics-Based Verification System for Flight Critical Software

Potential Applications

We anticipate that our tool and definition of CIL will be used in verification of all safety and mission critical flight systems. NASA has shown themselves to be very amenable to formal methods techniques in the past. Our technical contact within NASA expects such a tool to be used on vehicles targeted at manned space missions as well as mission critical systems such as the rockets, probes, space telescopes, and landers/rovers.